February, 2013

Technical Report: Record Linkage Between Montana Highway Patrol And Montana Vital Statistics Data, 2008-2011

Cody Custis, MS, Epidemiologist, MHDDS
Hannah Yang, BS, MSPH student, Johns Hopkins Bloomberg School of Public Health

Purpose And Background

Since 2008, the Montana Office of Vital Statistics (OVS) has received motor vehicle crash fatality report data from the Montana Highway Patrol (MHP). This makes it possible to use the MHP fatality crash data in conjunction with death record data maintained at OVS. Death certificates contain limited information about deaths caused by motor vehicle crashes; there are check boxes for accidental death and whether the deceased was the operator or passenger of a vehicle or a pedestrian if the death was due to a motor vehicle traffic accident, and 255 characters are available for the certifier to describe how the death occurred.[1] In contrast, MHP reports contain extensive information about fatal motor vehicle crashes, including location, type of vehicle, use of seat belts, information about road conditions, and alcohol involvement.

Before MHP reports can be used to expand information about motor vehicle crash fatalities, MHP reports must first be linked to death certificates. MHP and OVS datasets have a number of variables in common, but no unique common identifiers, such as Social Security number, to link individual records in the two datasets together. In the absence of a unique identifying variable, we developed a record linkage algorithm to link individual records from the two datasets. Vital statistics previously linked a list of deaths from California to a list of births from Montana using SAS software.[2] We have now modified and expanded the algorithm to be easy to understand, easy to code, ready to link a variety of linking variables, and possible to accommodate different datasets. Because the source code is written in a familiar programming language (SAS), we could readily design and modify the linkage process.

Because the MHP investigates motor vehicle crash fatalities that occur in Montana, we linked MHP data to death certificate data for deaths that occurred in Montana, regardless of residence of decedents, as long as the death was classified as a motor vehicle traffic fatality (including cyclists and pedestrians struck by motor vehicles) using the External Cause of Injury Mortality Matrix.[3] At the time of analysis, 2011 was the latest year for which complete records from OVS and MHP. The number of motor vehicle deaths ascertained from OVS was 875 and fatalities from MHP crash reports was 848.

Selecting Variables To Include In Matching Strategy

We selected date of injury as a blocking variable, that is, a variable which we require to match between linked records.[4] Blocking variables reduce the number of potential matches for consideration. Without a blocking variable, the algorithm would consider all 742,000 (875*848) potential matches. Date of injury is a

[1] http://www.cdc.gov/nchs/data/dvs/DEATH11-03final-ACC.pdf;
[2] Version 9.3 of the SAS System for Windows. SAS Institute Inc.
[3] www.icd10data.com; http://www.cdc.gov/nchs/data/ice/icd10_transcode.pdf
[4] www.cste.org/webpdfs/**linkage**conceptsrev.pptx

good blocking variable because it is rarely missing (present on 868 of 875 OVS deaths and every MHP fatality report), numeric (no need to handle spelling variations), and likely to be accurately recorded on both OVS and MHP datasets. We excluded the seven OVS records without date of injury from the record linkage algorithm and further analysis. Because we were unsure if medical examiners recorded the date of injury in the same manner as the MHP, we allowed date of injury to be off by one day between OVS and MHP datasets. For example, if OVS records the date of injury as March 20, MHP fatalities with date recorded on March 19, 20, or 21 are candidates for a match. This allows for more potential true matches at the cost of additional computation for the additional potential matches that need to be considered.

For pairs of records where the blocking variable, date of injury, matches, we included the year, month, and day of month of birth; first, middle, and last name; sex; and street address and zip code as separate linking variables. For each linking variable, the algorithm compares the variables between the two datasets. For example, if the first name variable from both an OVS and MHP record is 'OLIN,' then first name matches.[5] On the other hand, if the OVS and MHP records have last names of 'WELLIVER' and 'SCOTT,' last name does not match.

A deterministic record linkage algorithm decides if a record pair is a true match based on a selection of linking variables (for example, consider a record pair a true match if last name and all components of date of birth match). In contrast, a probabilistic record linkage algorithm assigns weights for all the linking variables based on the probability of agreement for true matched pairs and false matched pairs.[6] In probabilistic record linkages, the criterion for inclusion as a linking variable is that a variable be more likely to agree on matches than unmatched pairs. Sex would have a near certain probability of agreement for true matched pairs, but would be expected to have a probability of agreement of roughly ½ for an unmatched pair. On the other hand, first name would have a high probability of agreement for true matched pairs but be less likely (though possible) to agree for unmatched pairs. Address is less likely to agree, both because address is a more complex string ('2450 New York Avenue') and also because individuals may use post office boxes or work address instead of residential address, yet is still unlikely to agree on unmatched pairs.

Our experience with previous record linkage projects found that cross links of name variables (such as first name to middle name or middle name to last name) slightly increase the proportion of matched records for linking vital statistics data, but greatly increase the complexity of match algorithms. We decided that the slight increase did not justify doubling the complexity. When date of birth is misclassified, it is often done so by either a single day (October 19, 1970 to October 18, 1970) or year (October 19, 1970 to October 19, 1971). Therefore, we consider the three date of birth components separately, as two components would still match.

We did some standardization of string variables based on previous experience linking vital statistics data. Name misspellings such as 'WELLIVIER' for 'WELLIVER' are common. Algorithms such as SOUNDEX assign alphanumeric codes to names, for example, 'W416' for both 'WELLIVIER' and 'WELLIVER.'[7] Past experience found SOUNDEX worked poorly on vital statistics data, mainly by creating too many false positive matches; 'WOLIPIERO' also has a SOUNDEX code of 'W416'.[8] Instead, for character strings, we used the COMPRESS function in SAS to remove embedded spaces, the UPCASE function to standardize capitalization, and the SUBSTR function to take the first four characters of the string. Thus, 'WELLIVER' becomes 'WELL' and '938 Smith Road' becomes '938S.'

[5] Example names and personal information are from http://www.fakenamegenerator.com/
[6] Fellegi IP, Sunter AB A Theory for Record Linkage, JASA 1969
[7] http://www2.sas.com/proceedings/sugi29/072-29.pdf
[8] http://rosettacode.org/wiki/Soundex

Steps In Linkage

Our first step estimates the probability of agreement for each of the linking variables, given an unmatched pair. To do this, we took the Cartesian production of 400 records (sampled with replacement) from both datasets using PROC SURVERYSELECT in SAS to create an unmatched dataset of 160,000 linked records. For each linking variable, we estimated the probability of agreement, $p_{(Agree|Unmatch)}$, as the proportion of unmatched records where the variables agree. Based on previous experience that having an estimated probability of zero creates computational problems, we created a floor for the estimated probability of agreement, $\frac{1}{N_u}$, where N_u is the number of unmatched records, so that the derived probability of matching was strictly between 0 and 1. We wrote a SAS macro that takes two input datasets and the selected linking variables and returns a single row dataset with the probability of agreement. This macro gave us the probabilities of agreement for the linking variables for unmatched records (Row 1 of Table 1).

The second step estimates the probability of agreement for matched records, $p_{(Agree|Match)}$. From past experience, performing a record linkage using a basic deterministic algorithm gives good estimates of the probability of agreement. For the deterministic algorithms, we required that either two of three birth date variables (year, month, and day) or two of three name variables (first, last, middle) match. Using two different sets of variables (name and date) allows us to estimate the probability of agreement for those deterministic variables, rather than force them as unwanted blocking variables. Using the deterministic algorithm, we matched 758 of 867 OVS records and 758 of 848 MHP records (87% and 89%).

Manual review of the matched records confirmed that the links were true matches. A manual review of the records that were unmatched found some record pairs that should be linked, some due to transpositions between first, middle, and last name, such as referring to 'Olin Foley Welliver' rather than 'Olin Welliver Foley.' Also common were misspellings such as 'Olein' and 'Olin' or changes in hyphenation patterns, such as 'Chelsea Chapman Dube' and 'Chelsea Chapman-Dube.' The probabilistic step is designed to add these additional true matches.

Using the records from the deterministic match, we estimate the probability of agreement for the linking variables, this time for matched records. As before, we created a computational limit such that the probability of matching can be at largest 1- $\frac{1}{N_m}$, where N_m is the number of records from the deterministic match to keep the derived probability strictly between 0 and 1. We were able to use the same macro as used in the first step to get the same probabilities, this time for the matched records (Row 2 of Table 1).

The third step uses the probabilities of agreement and disagreement to estimate appropriate weights for agreement and disagreement.[9] The weight for agreement is $\log_2(\frac{P_{Agree|Match}}{P_{Agree|Unmatch}})$. The weight for disagreement is $\log_2(\frac{P_{Disagree|Match}}{P_{Disagree|Unmatch}}) = \log_2(\frac{1-P_{Agree|Match}}{1-P_{Agree|Unmatch}})$. We developed a SAS macro that takes the variables from the datasets created in steps one and two, and returns a dataset with the appropriate weights for agreement and disagreement (Rows 3 and 4 of Table 1).

[9] Herzog TN, Scheuren FJ, Winkler, WE Data Quality and Record Linkage Techniques, Springer 2007 97-101

Table 1. Estimated Probability Of Agreement For Matched And Unmatched Records

	Year of Birth	Month Of Birth	Day Of Birth	Last Name	First Name	Middle Name	Sex	Address	Zip Code
P(Agree|Unmatch)	0.016	0.081	0.032	0.002	0.005	0.022	0.545	0.004	0.016
P(Agree|Match)	0.972	0.987	0.972	0.982	0.984	0.600	0.996	0.524	0.582
Matched Weight	5.9	3.6	4.9	8.8	7.5	4.7	0.9	7.2	5.1
Unmatched Weight	-5.1	-6.1	-5.1	-5.8	-6.0	-1.3	-6.8	-1.1	-1.2

The fourth step does a probabilistic record linkage using estimated weights to rejoin the OVS and MHP data and assign a match probability based on the sum of the weights and agreement / disagreement of the linking variables. As in the deterministic record linkage, date of injury is a blocking variable. A full description of how to calculate the match probability based on agreement / disagreement can be found in Appendix 1.

Cautions Regarding Use Of This Method

The probabilistic record linkage considers all record pairs where the blocking variables match. For those pairs, it calculates a match probability based on which linking variables match, and the probability of those linking variables matching. If the match probability is greater than or equal to zero, the algorithm retains the record pair as a potential match. If there is more than one potential match, the algorithm selects the record pair with the highest match probability for each death record to be the true match. Using the probabilistic algorithm, we match 790 of 867 OVS records and 790 of 848 MHP records (91% and 93%, respectively). Manual review of the 32 new matches reveals that they are true matches, but were not matched in the deterministic step because of misspellings and transposition of names.

Although OVS records have extensive edit programs for completeness, MHP records are less complete. Of the 57 MHP records that were unlinked, 23 had no middle name and an additional seven had only a middle initial. Ten MHP records had no date of birth, and fifteen had either a blank address or an unknown address.

Because our goal was to design an easy-to-use algorithm for record linkage, we ignored some recommendations for record linkage. Our algorithm does not weigh the frequency of a name. More complex algorithms give greater weight to matching on uncommon strings, such as matching first name of 'DAMARIS' having greater weight than matching first name of 'SOPHIA.'[10] When examining communities where a particular name is very common, such as 'Singh' or 'Kaur,' modifying the algorithm may provide lower misclassification rates.[11]

Our match probabilities do not account for the lack of independence between linking variables. For example, if date of birth is missing on a record, then all three components of the date will disagree. Our algorithm uses SAS's default handling of missing values, which is that a value of missing on the first record matched a value of missing on the second record.

We did not do extensive record cleaning and standardization such as address parsing.[12] Adding record cleaning and standardization would make our SAS program more complex and past experience found that the minimal string standardization worked well for matching OVS records. We used very basic cutoffs for determining if record pairs were true matches, using a more advanced cutoff system with true, false, and

[10] http://www.ssa.gov/oact/babynames/

[11] http://www.thestar.com/news/article/240030

[12] Winkler WE The State Of Record Linage And Current Research Problems, Statistical Research Division, US Census Bureau 1999

possible (requiring manual review) record pairs allows researchers to have more control over misclassification rates.[13]

Conclusions

Deaths from drunk driving and failure to wear seat belts are preventable. MHP extensively investigates the circumstances of motor vehicle crashes, whereas OVS registers the death but usually lacks details such as alcohol use in the causal death chain when a decedent is killed by a drunk driver. In addition, seat belt use is not recorded on death certificates. Policy makers have a more accurate picture of the benefits of policy changes for drunk driving and seat belt use by combining OVS and MHP data.

For information about Montana Vital Statistics, please contact Bruce Schwartz, Vital Statistics Epidemiologist, Office of Epidemiology and Scientific Support, (406) 444-1756 or bschwartz@mt.gov
This document was published in electronic form only. Alternative formats of this document will be provided on request.
Please visit our website at http://www.dphhs.mt.gov/publichealth/epidemiology/index.shtml

[13] Fellegi IP, Sunter AB A Theory for Record Linkage, JASA 1969

Appendix 1. Calculating the match probability

For each linking variable in a record pair, we note if each linking variable matches. In the demonstration, birth year matches for the pair. Thus, it contributes a value of 5.9 (the matched weight) to the match probability. Address fails to match; it contributes -1.1 (the unmatched weight) to the match probability.

Table 2. Weight Calculations For Two Variables In A Match

	OVS	MHP	Match	Weight (Matched)	Weight (Unmatched)	Weighted Probability
Birth Year	1982	1982	1	6.0	-5.2	5.9
Address	3984	100S	0	7.2	-1.1	-1.1

We sum the contributions to get an overall match probability. In the example, year of birth, month of birth, day of birth, last name, first name and sex all match; the weighted contribution is positive for those variables. Middle name, address, and zip code do not match; those make a negative contribution. The match probability, as calculated in our algorithm, of 28.1 is an ordinal measure of the quality of this potential record pair. For the same death record, if one record pair has a match probability of 28.1, and another has match probability of 36.4, the record pair with the higher match probability should be selected.

Table 3. Total Weight

Variable	Year of Birth	Month Of Birth	Day Of Birth	Last Name	First Name	Middle Name	Sex	Address	Zip Code	Total
Match	1	1	1	1	1	0	1	0	0	
Weight (Matched)	5.9	3.6	4.9	8.8	7.5	4.7	0.9	7.2	5.1	
Weight (Unmatched)	-5.1	-6.1	-5.1	-5.8	-6.0	-1.3	-6.8	-1.1	-1.2	
Weighted Probability	5.9	3.6	4.9	8.8	7.5	-1.3	0.9	-1.1	-1.2	28.1

Appendix 2. SAS Code

```
/*Vital_Statistics_Linking_Macro.sas*/

/*************************************************************************/
/*Step 1. Link using a random select of the first and second datasets.*/
;
/*PROC SURVEYSELECT is used to create random samples for creating unmatched records pairs.*/
proc surveyselect data = WORK.MHPDEATHS method = URS rep = 1
                        sampsize = 400 seed = 23863 out = WORK.RANDHIGHWAY  outhits NOPRINT;
  id _all_;
run;
;
proc surveyselect data = WORK.MVDEATHS method = URS rep = 1
                        sampsize = 400 seed = 85136 out = WORK.RANDMVDEATHS outhits NOPRINT;
  id _all_;
run;
;
/*Complies a macro that creates an output dataset with a single row of estimated unmatched
probabilities.*/
/*Includes macro variables for input datasets, output dataset name, prefix for the variables that
contain the probabilities*/
/*and a clause used to join the variables (needed for estimating matched probabilities in later step.)*/
%MACRO CREATELINKPROPS(DATASETONEIN=%STR(RANDMVDEATHS), DATASETTWOIN=%STR(RANDHIGHWAY),
PROPDATAOUT=%STR(RANPROPS), PREFIXDATAOUT=%STR(U),
FROMCLAUSE = %STR(FROM &DATASETONEIN AS uno, &DATASETTWOIN AS dos ) );
PROC SQL OUTOBS = 160000 STIMER;
CREATE TABLE WORK.RANMATCH AS
SELECT
1/ COUNT(*) AS MINPROP,
YEAR(uno.DBIRDT) = YEAR(dos.MHPDOBSAS) AS INDBIRYEAR,
MONTH(uno.DBIRDT) = MONTH(dos.MHPDOBSAS) AS INDBIRMONTH,
DAY(uno.DBIRDT) = DAY(dos.MHPDOBSAS) AS INDBIRDAY,
SUBSTR(COMPRESS(uno.DLAST ),1,4)=SUBSTR(COMPRESS(UPCASE(dos.MHPLAST)),1,4)  AS INDNAMLAS,
SUBSTR(COMPRESS(uno.DFIRST),1,4)=SUBSTR(COMPRESS(UPCASE(dos.MHPFIR)),1,4)   AS INDNAMFIR,
SUBSTR(COMPRESS(uno.DMIDL ),1,4)=SUBSTR(COMPRESS(UPCASE(dos.MHPMID)),1,4)   AS INDNAMMID,
uno.DSEX=dos.MHPSEX AS INDSEX,
SUBSTR(uno.DADDRES1,1,4) = SUBSTR(dos.MHPSTR,1,4) AS INDADD,
INPUT(uno.DZIPA,5.) = dos.MHPZIP AS INDZIP
&FROMCLAUSE
;

CREATE TABLE &PROPDATAOUT AS
SELECT DISTINCT
MAX(MINPROP,MEAN(INDBIRYEAR))  AS &PREFIXDATAOUT.BIRYEARPROP,
MAX(MINPROP,MEAN(INDBIRMONTH)) AS &PREFIXDATAOUT.BIRMONTHPROP,
MAX(MINPROP,MEAN(INDBIRDAY))   AS &PREFIXDATAOUT.BIRDAYPROP,
MAX(MINPROP,MEAN(INDNAMLAS))   AS &PREFIXDATAOUT.NAMLASPROP,
MAX(MINPROP,MEAN(INDNAMFIR))   AS &PREFIXDATAOUT.NAMFIRPROP,
MAX(MINPROP,MEAN(INDNAMMID))   AS &PREFIXDATAOUT.NAMMIDPROP,
MAX(MINPROP,MEAN(INDSEX))   AS &PREFIXDATAOUT.SEXPROP,
MAX(MINPROP,MEAN(INDADD))   AS &PREFIXDATAOUT.ADDPROP,
MAX(MINPROP,MEAN(INDZIP))   AS &PREFIXDATAOUT.ZIPPROP
FROM WORK.RANMATCH
;

QUIT;
%MEND;
;
/*Executes created macro*/
%CREATELINKPROPS(DATASETONEIN=%STR(RANDMVDEATHS), DATASETTWOIN=%STR(RANDHIGHWAY),
PROPDATAOUT=%STR(RANPROPS), PREFIXDATAOUT=%STR(U))
;
/*Step 2. Link using a deterministic match*/
;
/*This step links the OVS and MHP databases using a deterministic match.*/
PROC SQL;
CREATE TABLE WORK.LINKED AS
/*We only need select the variables that will be used for linking in this step, since the goal is
estimate the probability of agreement for linking.*/
```

```
SELECT uno.DFILENO, DEATHY4, DINJDTG FORMAT = DATE10., DLAST, DFIRST, DMIDL, DBIRDT FORMAT = DATE10.,
DCAUSEC  ,
DSEX, DADDRES1, DZIPA,
dos.*
FROM WORK.MVDEATHS AS uno INNER JOIN WORK.MHPDEATHS AS dos ON
/*The blocking varibale.*/
(ABS(uno.DINJDTG - CDATESAS) <= 1) AND
/*At least two of three of both name and birth date variables must match.*/
(SUM(YEAR(uno.DBIRDT) = YEAR(dos.MHPDOBSAS), MONTH(uno.DBIRDT) = MONTH(dos.MHPDOBSAS), DAY(uno.DBIRDT) =
DAY(dos.MHPDOBSAS) ) >= 2 ) AND
(SUM(SUBSTR(COMPRESS( DLAST),1,4) =SUBSTR(COMPRESS(UPCASE(MHPLAST)),1,4),
     SUBSTR(COMPRESS(DFIRST),1,4) =SUBSTR(COMPRESS(UPCASE(MHPFIR )),1,4), SUBSTR(COMPRESS(DMIDL),1,4)
=SUBSTR(COMPRESS(MHPMID),1,4)
) >= 2)
;
QUIT;
;
/*Can execute macro again, only need to change prefix, input datasets, and joiningg clause.*/
/*The macro will give an error due to &DATASETONE not resolving at compilation.  It does resolve at
execution, and can be ignored.*/
%CREATELINKPROPS(DATASETONEIN=%STR(WORK.LINKED), DATASETTWOIN=%STR(WORK.LINKED),
PROPDATAOUT=%STR(MATPROPS), PREFIXDATAOUT=%STR(M),
FROMCLAUSE = %STR(FROM &DATASETONEIN AS uno INNER JOIN &DATASETTWOIN AS dos ON uno.DFILENO = dos.DFILENO
AND uno.DEATHY4 = dos.DEATHY4) )
;
/*Step 3.*/
/*Set up weights from the matched and unmatched indicators.*/
;
/*Compile macro to do arithmetic on matched and unmatched probabilities, shortens program.*/
%MACRO VARPATTERN(MATCHEDPREFIX=M,UNMATCHEDPREFIX=U,VARIABLENAME=BIRYEAR);
LOG2(&MATCHEDPREFIX.&VARIABLENAME.PROP/&UNMATCHEDPREFIX.&VARIABLENAME.PROP ) AS M&VARIABLENAME.WGT,
LOG2((1-&MATCHEDPREFIX.&VARIABLENAME.PROP)/(1-&UNMATCHEDPREFIX.&VARIABLENAME.PROP) ) AS
U&VARIABLENAME.WGT
%MEND;
;
/*Weighted created for all linking variables.*/
PROC SQL;
CREATE TABLE WORK.LINKINGWEIGHTSET AS
SELECT
%VARPATTERN(VARIABLENAME=BIRYEAR), %VARPATTERN(VARIABLENAME=BIRMONTH), %VARPATTERN(VARIABLENAME=BIRDAY),
%VARPATTERN(VARIABLENAME=NAMLAS),  %VARPATTERN(VARIABLENAME=NAMFIR),   %VARPATTERN(VARIABLENAME=NAMMID),
%VARPATTERN(VARIABLENAME=SEX),  %VARPATTERN(VARIABLENAME=ADD),   %VARPATTERN(VARIABLENAME=ZIP)

FROM WORK.RANPROPS, MATPROPS
;
QUIT;
;
/*Step 4.*/
/*Do record linkage again, using the weights calculated in the previous steps.*/
;
%MACRO INDWEIGHT(VARNAME=,MATCHEDPREFIX=M,UNMATCHEDPREFIX=U);
(calculated IND&VARNAME.*M&VARNAME.WGT)+( (1- calculated IND&VARNAME.)*U&VARNAME.WGT)
%MEND;
;
PROC SQL STIMER;
CREATE TABLE WORK.HIGHWAYLINK AS
SELECT *,
/*Indicator variables for agreement.*/
YEAR(uno.DBIRDT) = YEAR(dos.MHPDOBSAS) AS INDBIRYEAR,
MONTH(uno.DBIRDT) = MONTH(dos.MHPDOBSAS) AS INDBIRMONTH,
DAY(uno.DBIRDT) = DAY(dos.MHPDOBSAS) AS INDBIRDAY,
SUBSTR(COMPRESS(uno.DLAST ),1,4)=SUBSTR(COMPRESS(UPCASE(dos.MHPLAST)),1,4)  AS INDNAMLAS,
SUBSTR(COMPRESS(uno.DFIRST),1,4)=SUBSTR(COMPRESS(UPCASE(dos.MHPFIR)),1,4)   AS INDNAMFIR,
SUBSTR(COMPRESS(uno.DMIDL ),1,4)=SUBSTR(COMPRESS(UPCASE(dos.MHPMID)),1,4)   AS INDNAMMID,
uno.DSEX=dos.MHPSEX AS INDSEX,
SUBSTR(uno.DADDRES1,1,4) = SUBSTR(dos.MHPSTR,1,4) AS INDADD,
INPUT(uno.DZIPA,5.) = dos.MHPZIP AS INDZIP,
/*Sum of indicator variables multiplied by the appropriate weight for each indicator.*/
SUM(
```

```
%INDWEIGHT(VARNAME=BIRYEAR), %INDWEIGHT(VARNAME=BIRMONTH), %INDWEIGHT(VARNAME=BIRDAY),
%INDWEIGHT(VARNAME=NAMLAS),  %INDWEIGHT(VARNAME=NAMFIR),   %INDWEIGHT(VARNAME=NAMMID),
%INDWEIGHT(VARNAME=SEX),  %INDWEIGHT(VARNAME=ADD),   %INDWEIGHT(VARNAME=ZIP)

) AS FELSUNTOTAL
FROM WORK.MVDEATHS AS uno FULL JOIN WORK.MHPDEATHS AS dos ON
ABS(uno.DINJDTG - dos.CDATESAS) <= 1,  WORK.LINKINGWEIGHTSET
/*Only keep record pairs where the estimated odds of matching are positive.*/
HAVING FELSUNTOTAL >= 0
ORDER BY DEATHY4, DFILENO, FELSUNTOTAL DESCENDING;
;
CREATE TABLE WORK.HIGHWAYBEST AS
/*Find the record pair with the highest odds of matching...*/
SELECT uno.*, MAX(FELSUNTOTAL) AS BESTTOTAL, dos.*
FROM WORK.HIGHWAYLINK AS uno NATURAL LEFT JOIN HIGHWAY.'CRASH INFO TABLE'n AS dos
/*for each death certificate number each year...*/
GROUP BY DEATHY4, DFILENO
/*and keep those record pairs.*/
HAVING FELSUNTOTAL = BESTTOTAL;
;
QUIT;
;
```